REMARKS

Claims 1-6, 8-16, 18-20, and 22-35 are currently pending in the subject application and are presently under consideration. Claims 1, 11, 20 and 32 have been amended as shown on pp. 2-6 of the Reply. Claim 2 has been canceled.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-6, 8-16, 18-20, and 22-35 Under 35 U.S.C. §103(a)

In the Final Office Action dated January 16, 2009, claims 1-6, 8-16, 18-20, and 22-35 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Cynerman (Automate your build process using Java and Ant) in view of Jerger *et al.* (US 6,321,334). Reversal of this rejection is requested for at least the following reasons. Cynerman and Jerger *et al.*, alone or in combination, fail to teach or suggest all the claimed aspects.

The claimed subject matter provides a mechanism called "sandboxing" of a build platform that allows a developer to safely download, use, and augment their build processes. In one implementation, sandboxing allows the developer to mark different build entities with different levels of trust thereby mitigating the need of developers to fully trust all processes. Toward that end, claim 1 (and similarly claim 11) recites *a build process processor that processes one or more build entities; and a policy component that is processed by the build process processor to determine one or more levels of trust within which the build process operates;* and ***wherein a developer associates the one or more build entities with one or more levels of trust, such that at build time, a principal permission level under which the build process executes is determined by analyzing the levels of trust associated with each of the build entities, and lowest level of trust of all involved build entities dictates the principal permission level for execution of the build process; and wherein the levels of trust include:***

- ***(i) levels that are representative of trusted, which has no restrictions to the build process,***
- ***(ii) semi-trusted, which has restrictions to the build process and***
- ***(iii) untrusted, which causes the build process to fail,***

wherein if the lowest level of trust is untrusted and the build process fails, the developer is notified. The cited art fails to teach or suggest such claimed aspects.

Cynerman teaches utilizing an 'Ant' tool to execute an automated build process. Ant facilitates constructing build scripts *via* a large number of built-in tasks without any customization. The 'simple.xml' is not a policy component that determines one or more levels of trust for the build process. Rather the file 'simple.xml' is an xml file with a project entity comprising several target entities wherein the first line has information about an overall project to be built with target tasks and related attributes (*See e.g.* Cynerman page 3). Further, 'if' and 'less' commands can be used in the simple.xml to specify commands that are to be performed either *if* a certain property is set or *unless* that property is set. The *if* will execute when the property value is set, and the *unless* will execute if the value is not set (*See e.g.* Cynerman page 4). However, it fails to teach or suggest setting up a policy that sets a level of trust by which a conditional build process is executed as conceded on page 5 of the Final Office Action dated January 16, 2009.

A secondary document, Jerger *et al.*, is cited to overcome this deficiency. Jerger *et al.* relates to a security model for managing foreign content downloaded from a computer network. It teaches associating security zones with network locations and configurable protected operations corresponding to these zones that control the access to the host system by foreign content downloaded from the computer network (*See e.g.* Jerger *et al.* Abstract). The operations corresponding to these security zones are executed based on defined permissions. However, it fails to make up for the aforementioned deficiency of Cynerman as it does not teach or suggest granting or denying permissions to specific build entities as recited in the subject claims. At the cited portion, Jerger *et al.* teaches editing of permission parameters within three permission sets associated with each security zone. Accordingly three permission sets are defined for signed and unsigned contents associated with different security zones and user interfaces are employed to allow a user to set permissions for different content from various security zones.

In view of the aforementioned, it is clear that the system of Jerger *et al.* requires voluntary input from the user in order to run a build process at a specific level of trust. In contrast, the claimed subject matter relates to a build process processor and policy component, wherein a developer associates build entities with a corresponding level of trust. Accordingly, the build process executes at a permission level that is the lowest of trust levels associated with the build entities. However, if any of the entities are deemed untrusted, the lowest trust level is deemed untrusted and the build process will abort with a security message, before the process even begins. By automatically selecting a lowest trust level from all the trust levels associated with the build

entities involved in the build process, the claimed subject matter mitigates a need for the user to specify trust levels for each entity as taught by Jerger *et al.* even while safely executing the build process.

Further, Jerger *et al.* merely discloses configuring multiple security zones, each security zone corresponding to a set of locations on a computer network. Each zone has a corresponding security configuration that specifies the actions to be taken when a protected operation is requested by active content downloaded from that security zone. An Internet security manager first determines the zone that the class was downloaded from when the active content is first downloaded. The downloaded class is then checked in a signature verification decision to determine if the downloaded class has been digitally signed. If not signed, then the Internet security manager retrieves the unsigned default permission set for the zone from which the class was downloaded. The Internet security manager then grants the permissions contained in the unsigned permission set. The permissions granted can either be temporary or permanent. (*See* col. 31, lines 47-67). Applicants' claimed subject matter associates build entities with one or more levels of trust. The levels of trust include trusted, which has no restrictions, semi-trusted, which has restrictions and untrusted which causes the build process to fail. Jerger *et al.* does not disclose levels of trust associated with build entities, such that the lowest level of trust of all involved build entities dictates the principal permission level for execution of the build process, as in applicants' claimed subject matter. Jerger *et al.* merely determines the zone that a class was downloaded from and applies the default permission set for that particular zone.

Furthermore, independent claim 20 recites a computer-readable medium having computer-executable instructions for performing a method for managing a build process, the method comprising *receiving a build process for building one or more build entities;* ***associating the one or more build entities with a level of trust, wherein the levels of trust include:***

(i) allowing any operation to be performed,

(ii) allowing only a minimal set of operations to be performed and

(iii) aborting the build process,

determining a principal permission level under which the build process executes by analyzing the levels of trust associated with each of the build entities; performing the build process at the lowest level of trust of all involved build entities; and if the lowest level of trust is aborting the build process, then notifying a user that the build process failed.

As stated *supra*, Cynerman teaches utilizing an 'Ant' tool to execute an automated build process. Ant facilitates constructing build scripts *via* a large number of built-in tasks without any customization. And, the 'simple.xml' is not a policy component that determines one or more levels of trust for the build process. Furthermore, Jerger *et al.* teaches editing of permission parameters within three permission sets associated with each security zone. Accordingly three permission sets are defined for signed and unsigned contents associated with different security zones and user interfaces are employed to allow a user to set permissions for different content from various security zones. Thus, Jerger *et al.* does not disclose levels of trust associated with build entities, such that the lowest level of trust of all involved build entities dictates the principal permission level for execution of the build process, as in applicants' claimed subject matter. Jerger *et al.* merely determines the zone that a class was downloaded from and applies the default permission set for that particular zone.

Further, independent claim 32 recites a system that facilitates control of a building process, comprising *means for providing an association between one or more build entities and a level of trust,* ***wherein the levels of trust include:***

(i) allowing any operation to be performed,

(ii) allowing only a minimal set of operations to be performed and

(iii) aborting the build process,

means for determining a principal permission level under which the build process executes by analyzing the levels of trust associated with each of the build entities; means for performing the build process at the lowest level of trust of all involved build entities used during the build process; and if the lowest level of trust is aborting the build process, then means for notifying a user that the build process failed.

As stated *supra*, Cynerman teaches utilizing an 'Ant' tool to execute an automated build process, wherein, the 'simple.xml' is not a policy component that determines one or more levels of trust for the build process. Furthermore, Jerger *et al.* teaches editing of permission parameters within three permission sets associated with each security zone. Thus, Jerger *et al.* does not disclose levels of trust associated with build entities, such that the lowest level of trust of all involved build entities dictates the principal permission level for execution of the build process, as in applicants' claimed subject matter. Jerger *et al.* merely determines the zone that a class was downloaded from and applies the default permission set for that particular zone. By automatically

selecting a lowest trust level from all the trust levels associated with the entities involved in the build process, the claimed subject matter mitigates a need for the user to specify trust levels for each entity as taught by Jerger *et al.* even while safely executing the build process.

In view of at least the aforementioned deficiencies, it can be concluded that Cynerman in view of Jerger *et al.* fails to teach or suggest all aspects recited in independent claims 1, 11, 20 and 32. Therefore, reversal of the rejection of these independent claims and claims dependent there from is respectfully requested.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP582US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,
TUROCY & WATSON, LLP

/Marisa J. Zink/
Marisa J. Zink
Reg. No. 48,064

TUROCY & WATSON, LLP
57TH Floor, Key Tower
127 Public Square
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731